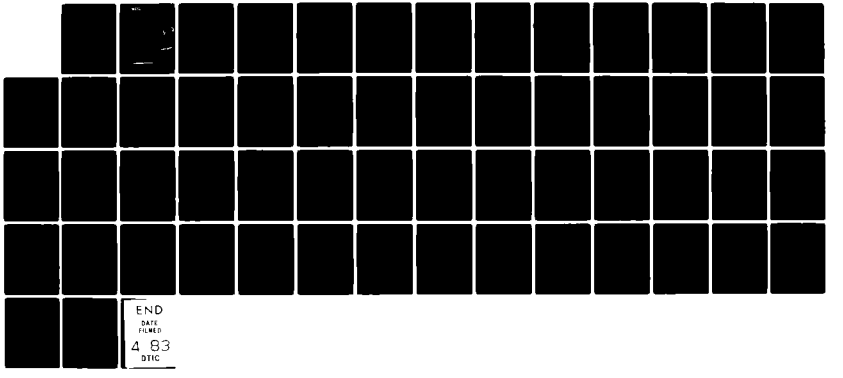
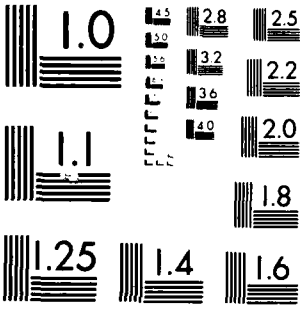
AD-A126 555 PROPOSED DOD THP STANDARD(U) SYSTEM DEVELOPMENT CORP 1/1
SANTA MONICA CA 02 DEC 81 SDC-TM-7038/211/00
DCA100-80-C-0044

UNCLASSIFIED F/G 17/2 NL



END
DATE FILMED
4 83
DTIC



SDC
System Development Corporation
2500 Colorado Avenue, Santa Monica, CA 90406, Telephone (213) 820-4111

series base no./vol./reissue
TM-7038/211/00
author
DTI Staff
technical Carl M. Switzky
Carl M. Switzky
release David J. Kaufman
David J. Kaufman
for
Charles A. Savant
date
12/2/81

TM a working paper

This document was produced by
System Development Corporation in performance of Contract DCA100-80-C-0044

DCEC PROTOCOLS STANDARDIZATION PROGRAM

PROPOSED DoD THP STANDARD

ABSTRACT

This document, prepared by Digital Technology Inc. under subcontract to System Development Corporation, is the initial service specification for the Terminal-to-Host Protocol, a virtual terminal protocol which resides in the presentation layer of the Department of Defense Architectural Model. In order to provide a framework for the understanding of the protocol, the virtual terminal model is described. The services provided by THP and those required from the underlying session layer are then specified. The THP services specified include basic services and specific enhancements to support scroll, paged, and forms classes of terminals.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 7038/211/00	2. GOVT ACCESSION NO. A126 553	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Proposed DoD THP Standard		5. TYPE OF REPORT & PERIOD COVERED interim technical report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) DTI Staff		8. CONTRACT OR GRANT NUMBER(s) DCA100-80-C-0044
9. PERFORMING ORGANIZATION NAME AND ADDRESS System Development Corporation 2500 Colorado Ave. Santa Monica, CA 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS P.E. 33126K Task 1053.558
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Engineering Center Switched Networks Engineering Directorate 1860 Wiehle Ave., Reston, VA 22090		12. REPORT DATE 2 Dec 81 13. NUMBER OF PAGES 51
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) N/A		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

This document represents results of interim studies which are continuing at the DCEC of DCA.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Protocols, Data Communications, Data Networks, Protocol Standardization, Terminal-to-Host Protocol.

20 ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document, prepared by Digital Technology Inc. under subcontract to System Development Corporation, is the initial service specification for the Terminal-to-Host Protocol, a virtual terminal protocol which resides in the presentation layer of the Department of Defense Architectural Model. In order to provide a framework for the understanding of the protocol, the virtual terminal model is described. The services provided by THP and those required from the underlying session layer are then specified. The THP services specified include basic services and specific enhancements to support scroll, paged, and forms classes of terminals.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

LIST OF FIGURES

1. OVERVIEW

1.1 INTRODUCTION

This report is the initial version of the Terminal-to-Host Protocol, Version 3 (THP-3). THP-3 is based on the virtual terminal approach. In this approach, a canonical or virtual terminal is defined. Input from a terminal on one host is mapped from the local terminal format into the virtual terminal format, which serves as the transfer syntax. The data is sent over the network to a remote host, which maps the virtual terminal format into that system's local format. Thus, to provide network terminal access, each host must support only one new terminal type, the virtual terminal.

THP-3 is a protocol of the presentation layer, which is described in the Department of Defense (DoD) Protocol Architecture Model [SYTEK, 1980] and in the Open System Interconnection Reference Model [ISO, 1981a]. As such THP-3 provides services to the application layer and receives services from the session layer.

This initial THP-3 standard specifies the virtual terminal model, the services provided by THP-3, and the services required from the underlying session layer. The THP-3 services defined include basic services and specific enhancements to support scroll, paged, and forms classes of terminals.

1.2 BACKGROUND

The virtual terminal approach has been used for some time. However, its application has been refined for general use only recently. The ARPANET Telnet protocol [ARPA, 1973] first used the technique. This protocol was very successful in supporting a simple scroll mode terminal. In 1976, the Stanford Research Institute designed the Terminal-to-Host Protocol (THP-1) [Postel, et al., 1976] for use in the then proposed AUTODIN II network. This protocol is very similar to Telnet, although slight changes were made to improve efficiency and to provide for DoD security and precedence requirements. In 1977, this Telnet-THP work was combined with virtual terminal research in Europe (such as described in Schicker and Duenki [1977]) to produce what is referred to as the INWG virtual terminal protocol [IFIP, 1978]. The International Network Working Group (INWG) is Working Group 6.1 of the International Federation for Information Processing (IFIP WG 6.1), under whose auspices the work was done. The INWG virtual terminal protocol was the first attempt to develop a virtual terminal model general enough to cover a wide range of terminal classes. The model included scroll, paged, and data entry terminal classes and provided for extension of services [Day, 1980].

Starting in 1978, the International Standards Organization (ISO) developed the Open System Interconnection Architecture (OSIA). OSIA clarified the position of terminal protocols in the overall architecture, as well as the precise services and functions that such protocols should offer. This led to work by ISO committee TC97/SC16/WG5 [ISO, 1980] and by a technical group of the European Computer Manufacturers Association (ECMA) [ECMA, 1981a, 1981b] to develop a virtual terminal service that fits into the OSIA framework.

In 1979, a revised version of THP-1 was produced by Computer Science Corporation [CSC, 1979]. This protocol, referred to here as THP-2, represented the conventions to be implemented in the AUTODIN II network. THP-2 modified some of the protocol mechanisms in THP-1 and added a few new options.

DoD applications require more sophisticated terminal capabilities than just scroll mode. Paged and forms mode terminals are widely used, and the number of graphics devices is on the rise. The architectures of THP-1 and THP-2 cannot efficiently support these more sophisticated devices. Therefore, this report, based heavily on the ECMA work, takes a more comprehensive approach to virtual terminal services. It lays the groundwork for a terminal protocol capable of supporting a much wider class of terminals. The protocol will be referred to as THP-3 and the service provided by the protocol as the virtual terminal service.

1.3 OVERVIEW OF THIS SPECIFICATION

This document specifies the services required to support four classes of terminals:

a. Basic Class

b. Scroll Class

c. Paged Class

d. Forms Class

The application of the model is limited to these classes of terminals because they are most immediately needed. These services must be provided first before moving to the more sophisticated services. The THP-3 is organized to be readily extensible, so other classes of service may be specified at a later time. The generic model and its extension mechanism help to ensure that new services can be developed in a consistent way.

The current ISO and ECMA work represents the most complete exposition of virtual terminal services available. While THP-3 conforms to some aspects of their work, such as modeling a display as a one, two, or three dimensional array, a number of aspects of the ISO and ECMA positions do not adequately reflect the environment in the DoD or in the U.S. Whenever THP-3 conforms to ISO and ECMA work, their text is used, distinguished by sidebars. For areas of disagreement, new text was developed, with no sidebars, to describe the necessary changes. These areas of disagreement will be used for a series of contributions to American National Standards Committees and ISO. The ECMA text was modified slightly to make it consistent with the changes. These modifications are enclosed in square brackets.

This report is divided into eight major sections.

1. Section 1 contains introductory material.

2. Section 2 describes the virtual terminal model that serves as the framework for this specification.

3. Section 3 describes the services THP-3 provides to the application layer.

4. Section 4 specifies the service interface to the application layer.

5. Section 5 describes the services required from the session layer.

6. Section 6 specifies the service interface with the session layer.

7. Section 7 specifies the protocol.

8. Section 8 specifies the services required from the local system environment.

Sections 1, 2, 3, and 5 are provided in this initial version. The remaining sections will be provided in a later version.

1.4 SCENARIO

The following scenario illustrates the operation of THP-3 by describing the general sequence of events that might be expected to occur for a simple terminal session. The description is stated in terms of the virtual terminal model and services that are elaborated more fully in Sections 2 and 3. The sequence of events portrayed is organized according to four phases of service: establishment, negotiation, data transfer, and termination.

The session described is purposely simple so that THP-3's operation is not obscured by the details of complex sessions with multiple connections or extensive use of options. However, such complex sessions may easily be conceptualized by compounding the simpler events depicted here.

Every effort has been made not to imply specific mechanisms in the scenario. Nonetheless, once THP-3 mechanisms are defined, it may be necessary to modify the scenario to conform to those mechanisms.

In the following scenario, a scrolling terminal supported by host A is to exchange data with a remote application supported by host B. The application has been designed to function with scrolling, paged, or forms terminals, and it permits the terminal interface to determine the class of service appropriate for a given terminal. The terminal's characteristics are compatible with a standard scroll class profile. The exchange proceeds as follows:

Establishment

a. To initiate the session, the terminal interface in host A requests establishment of a presentation connection to support THP-3 operations with the application in host B (see Figure 1). The terminal interface may be designed to initiate such activities in response to commands generated at the terminal or within system management modules of host A.

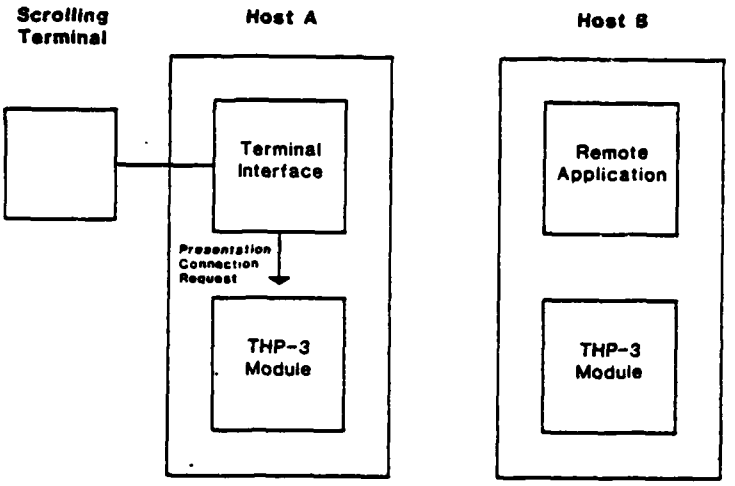


Figure 1. Initiation of Virtual Terminal Services

b. The THP-3 module in host A records the request data and attempts to satisfy the request by establishing a suitable session connection (see Figure 2). It also maintains the status of the pending request until success or failure of session connection establishment is determined. The presentation connection request data may be conveyed along with the session connection request. This permits efficient request processing by minimizing the communications overhead of connection establishment protocol exchanges. The details of session layer operations are beyond the scope of this report. However, as depicted in the figure, the session layer protocol modules employ services of lower layer protocol modules to establish a session connection.

c. Having established a session connection, the remainder of the presentation connection process proceeds as in Figure 3. The presentation connection request data is conveyed to the remote THP-3 module in host B. The THP-3 module generates a connection request, sends it to the remote application, and conveys the application's response back to the initiating side.

d. On receiving a response, the THP-3 module in host A resolves the pending presentation connection request by generating a response for the terminal interface. This response indicates success or a reason for failure to establish the requested connection. Assuming that the connection is successfully established, the situation in Figure 4 results. The applications may communicate on a virtual

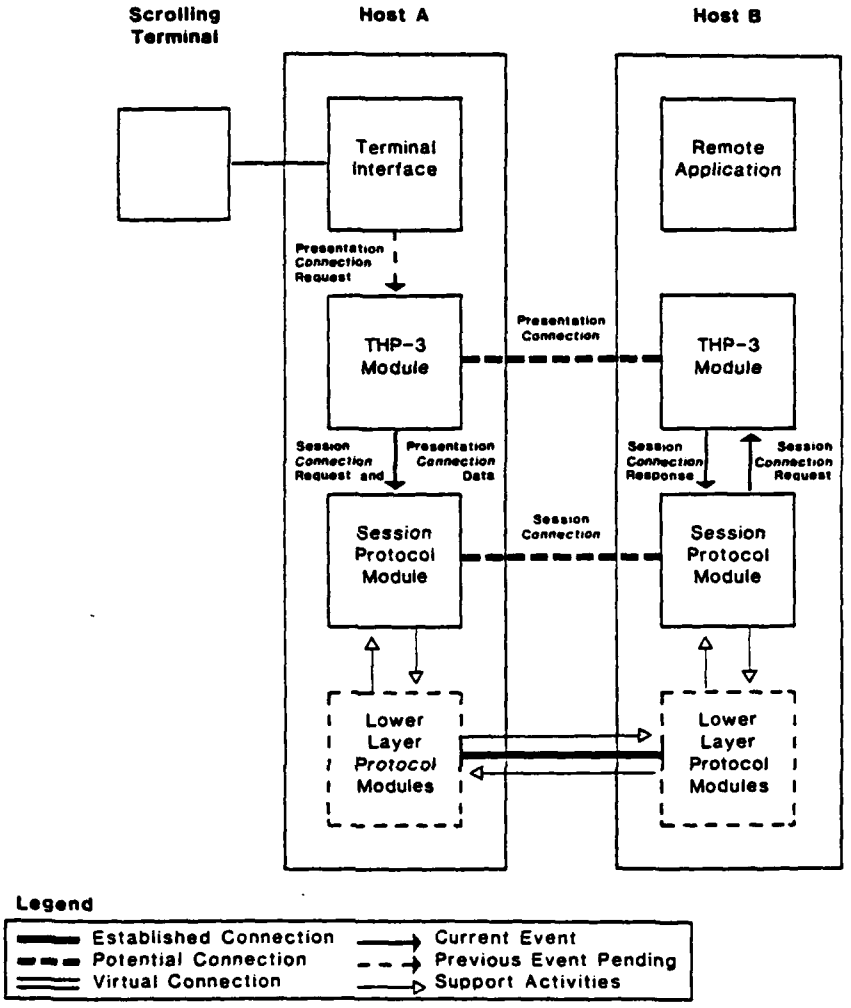


Figure 2. Session Connection Establishment

connection supported by THP-3 services supported by the established presentation connection. The presentation connection is in turn supported by session layer services.

Negotiation

a. By default, THP-3 supports virtual terminal services based on the simple basic class service parameters. Assume that the terminal interface defers negotiation of alternative service parameters to the addressed application. Suppose that the remote application negotiates the terminal class by offering a choice of forms, paged,

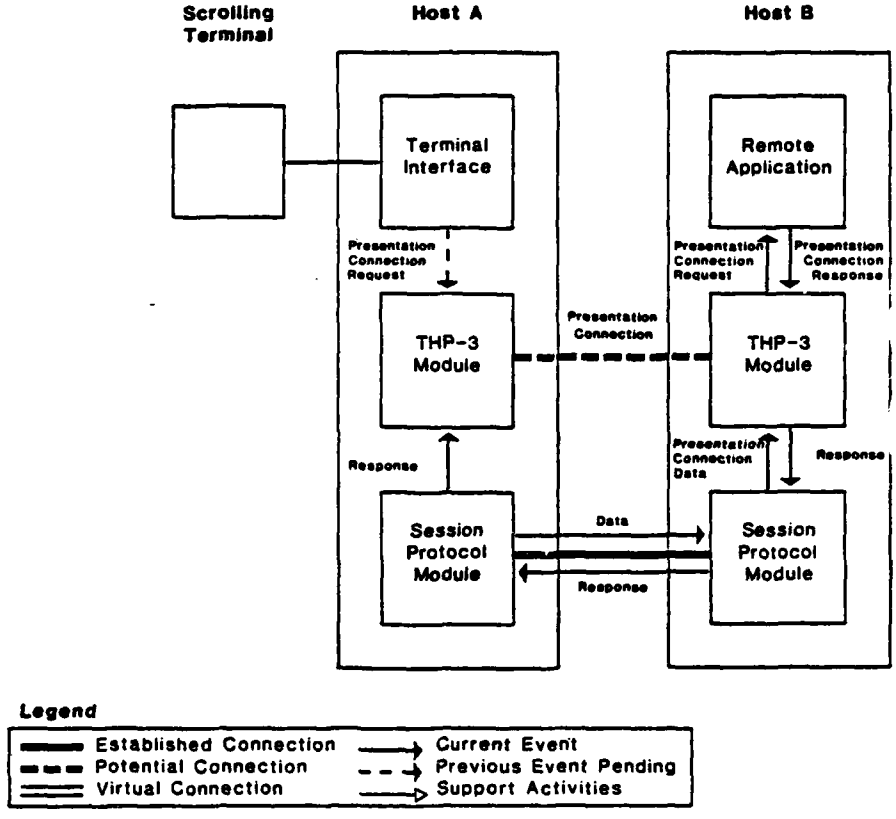


Figure 3. Presentation Connection Establishment

or scroll class services (see Figure 5). THP-3 may check that such negotiation events are consistent with pending negotiations. In this example, there are no pending negotiations, so the THP-3 module in host B merely relays the offer and maintains its status until a response or connection termination request is received.

b. The peer THP-3 module in host A also checks negotiation events for consistency and maintains their statuses. There are no pending negotiations, so it informs the terminal interface of the offer.

c. The terminal interface then selects the appropriate class for the terminal to be used. It may reject the offer or accept by selecting one of the alternatives. Assume that the scroll class is selected. The THP-3 module conveys the response to its peer and updates the negotiation status and service parameters. Thus, the default profile of the scroll class becomes effective for the terminal side of the connection.

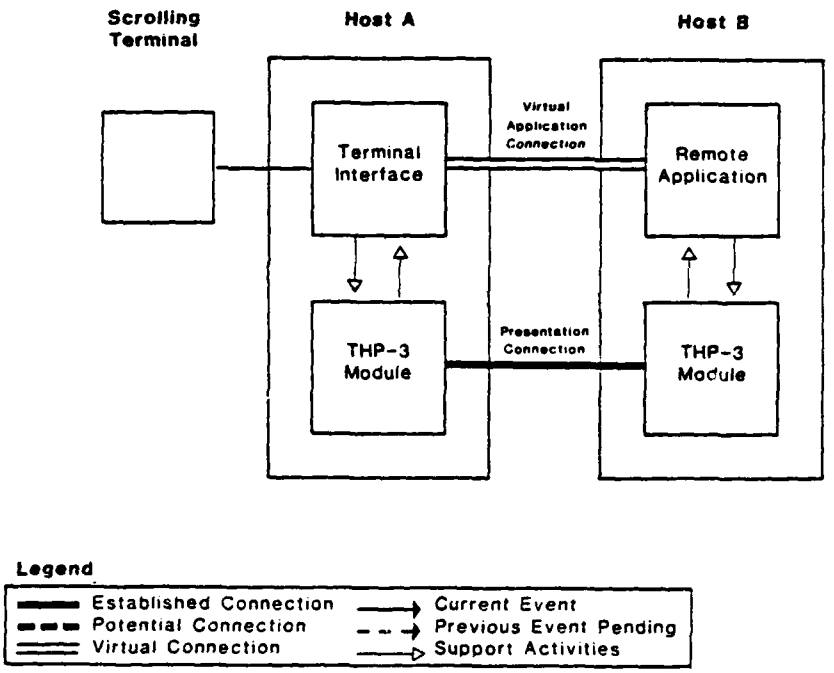


Figure 4. Established Presentation Connection

d. The peer THP-3 module in host B subsequently conveys the response to the application and updates its status and parameter values.

e. A variety of other negotiation schemes may be supported by THP-3 services. To illustrate one of these, assume that the application defers its privilege to control the course of negotiations to the terminal interface. This convention must be incorporated into both the terminal interface and the remote application module to be effective. The invitation is conveyed to the THP-3 module in host B, which then relays it to the peer THP-3 module in host A. The THP-3 module in host A passes it to the terminal interface (see Figure 6).

f. The terminal interface uses the second round of negotiations to establish a non-default scroll class profile for the session (see Figure 7). Instead of offering a choice of profiles, the terminal interface simply selects the appropriate profile and conveys its selection to the THP-3 module. The THP-3 module proceeds as in the negotiation of the terminal class and conveys the selection to its peer.

g. The peer THP-3 module in host B indicates the selection to the application, which may then either accept or reject it. If it is

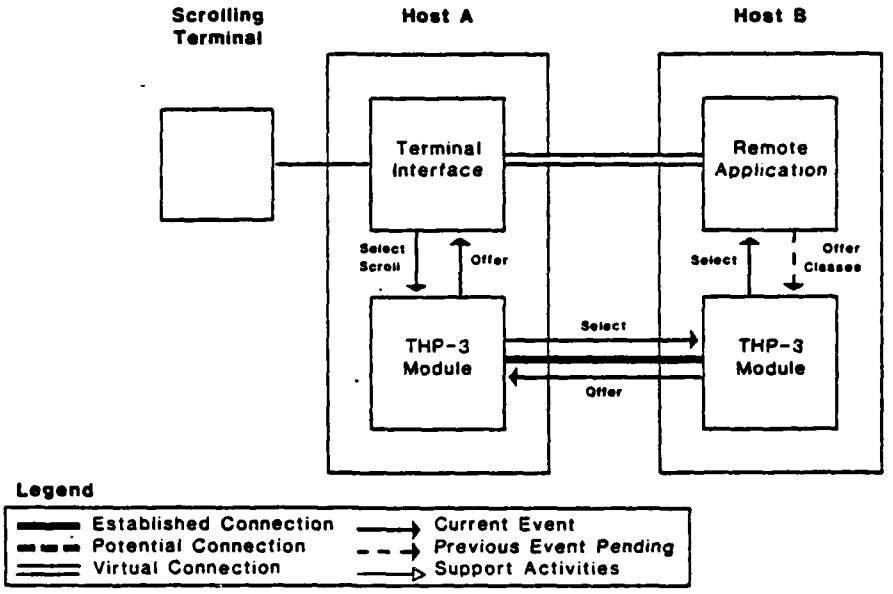


Figure 5. Selection from Offered Alternatives

accepted, the positive response is conveyed back, and each THP-3 module in turn updates the service parameters to reflect the change. Finally, the terminal interface is informed of the acceptance of its selection.

Data Transfer

a. Having determined values for the service parameters, the application and the terminal interface may now exchange data. Assume that the remote application in host B initiates the exchange. It sends a series of controls to initialize the terminal and a few lines of text to introduce the application and prompt the terminal operator for input. Each transfer is initiated by a request to the THP-3 module in host B (see Figure 8), which conveys the data to its peer for submission to the terminal interface.

b. For each transfer, the THP-3 module accepts data in the local syntax of host B and translates it into the virtual terminal syntax for transmission. It also implements all aspects of access and transfer control. In this example, only echo control is assumed to be relevant. Assume that local echoing is one of the parameters of the profile negotiated for the session. This applies to the terminal side of the connection, so the application side need effect no additional controls on the transferred data.

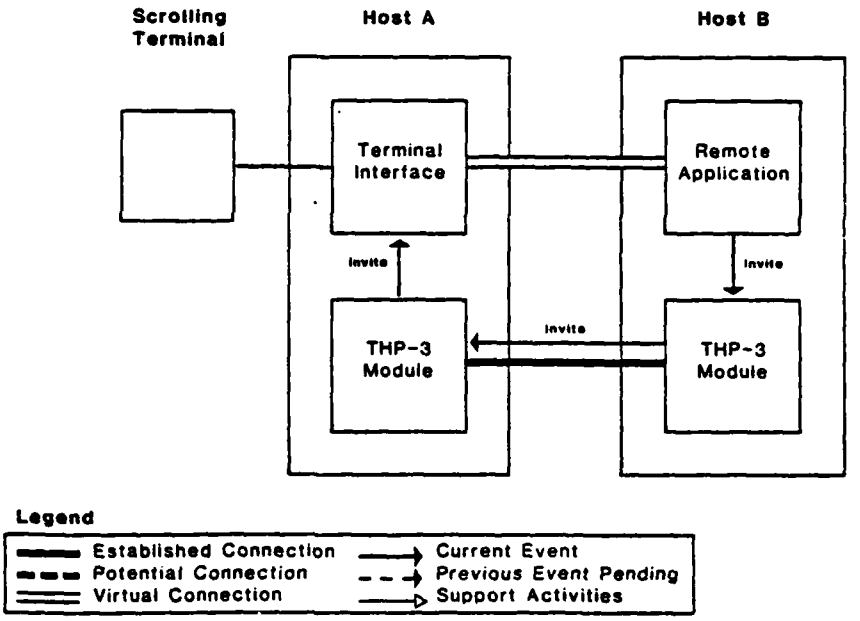


Figure 6. Invitation to Make Offer

c. The peer THP-3 module in host A translates the data from the virtual terminal syntax into the local syntax of host A and delivers it to the terminal interface. The terminal interface issues appropriate commands to the terminal.

d. Responses are accomplished in much the same way. Local echoing for the terminal may easily be accomplished within the terminal interface or other local system software, so the THP-3 module in host A need not perform any special processing.

Termination

a. After some exchange of data the application determines that the goals of the session have been attained. It then initiates activities to terminate the connection by conveying a termination request to the THP-3 module (see Figure 9). The THP-3 module relays the request to its peer in host A and maintains the status of the pending request until it receives a response or connection abortion indication. It will also suspend processing of additional requests from the application until normal status is regained, if ever.

b. The peer THP-3 module similarly marks the receipt of the request and relays it to the terminal interface. It does not necessarily suspend other request processing for the terminal interface.

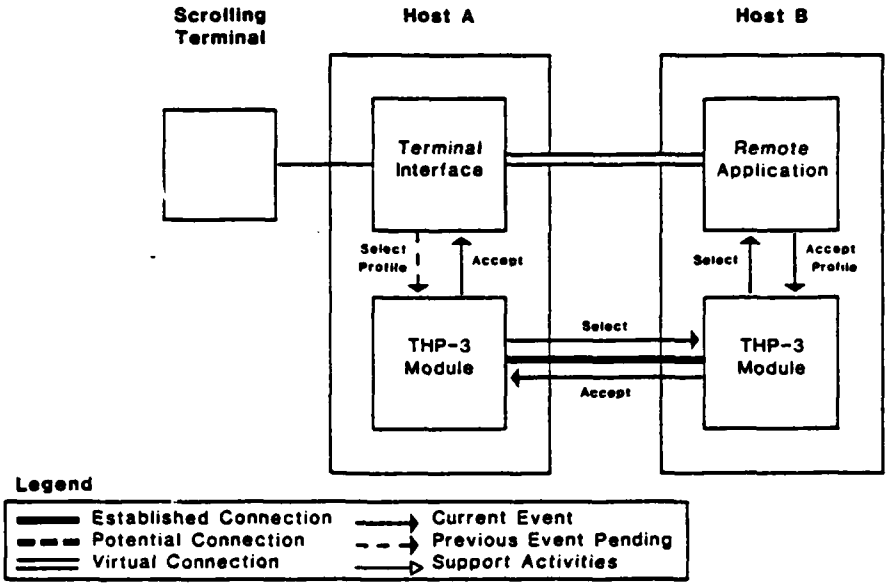


Figure 7. Acceptance of Alternatives Selected

c. If a positive response is received, as is assumed here, the THP-3 module in host A discontinues services for the terminal interface and conveys the response to its peer. The peer relays the response to the application and ceases services for it. Each THP-3 module in turn abolishes the presentation connection by issuing a termination command to the session protocol modules (see Figure 10).

If extreme conditions had arisen during the session that precluded continuation of THP-3 services, several abnormal termination schemes could have been initiated. The initial indication of such a condition could arise from a variety of sources (see Figure 11).

The primary constraint on the operation of the THP-3 modules is that an indication of the abortion be conveyed to each application, peer, or session module as appropriate. Thus, if the session module on the terminal side initiates the abortion by indicating this intent to its peer session module and to the THP-3 module in host A, the THP-3 module must relay the indication to the terminal interface. The session module in host B would then notify the THP-3 module in host B, which would indicate the abortion to the remote application.

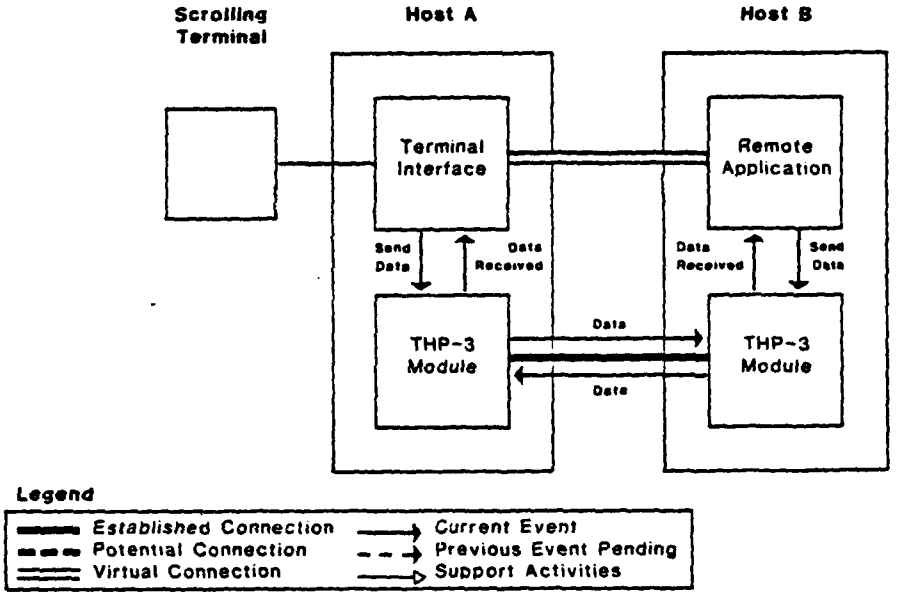


Figure 8. Data Transfer

2. VIRTUAL TERMINAL MODEL

2.1 OVERVIEW

This section considers the internal organization of the virtual terminal in greater detail. The virtual terminal model provides a conceptual framework for specifying the virtual terminal services and protocol mechanisms. The model is intended to be sufficiently general to apply to the terminal classes defined here and in later extensions. As noted in Section 1.3, text taken directly from ISO and ECMA is distinguished by sidebars.

Primarily the model consists of a (single) Conceptual Communication Area (CCA) that contains all the necessary information to allow the communicating partners (application(s) and/or terminal(s)) to derive a consistent view of the Virtual Device(s) that comprise the Virtual Terminal. The CCA is a conceptual repository for all information sent between the peers. The CCA will not have a physical existence, but each partner will normally have its own realization of it.

As shown in Figure 12, the CCA contains the following component information structures:

a. Data Structure Definition (DSD)

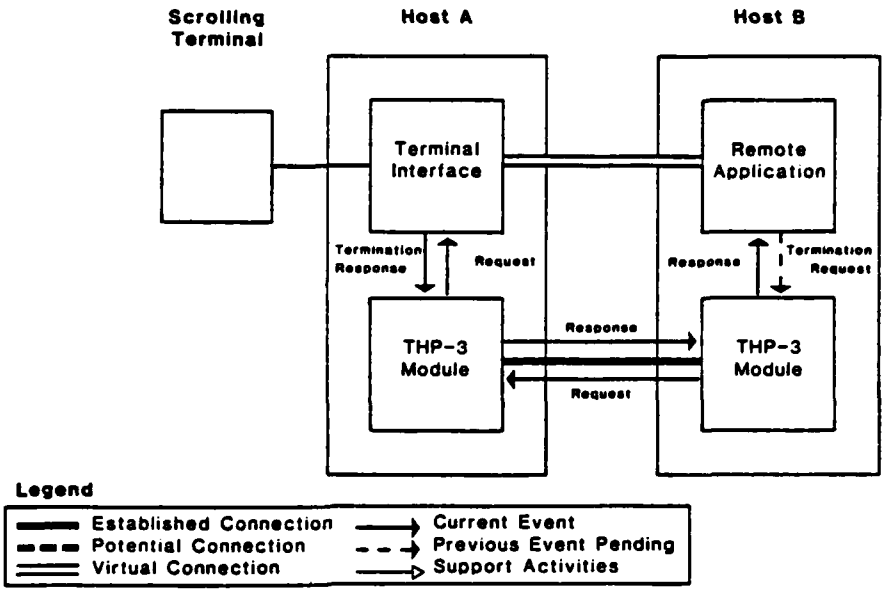


Figure 9. Initiation of Normal Termination

b. Conceptual Data Store (CDS)

c. Access Control (AC)

d. Signal Data Store (SIDS)

e. Status Data Store (SADS)

f. Presentation Interpreter (PrI)

g. Protocol Interpreter (PI)

Using the above information structures and their contents stored in the CCA, each partner is able to construct the state of its local Virtual Terminal. Such a construction implies the following:

a. the partner is able to visualize an image of the data content of the CDS; this image is referred to as the Conceptual Image, which may be routed to any of the virtual devices.

b. the partner is able to modify, in an orderly manner, the content of the CDS, including the attribute information, if present.

c. the partner is able to control or interpret the status and content of Signalling Mechanisms of the Virtual Terminal, where such are available.

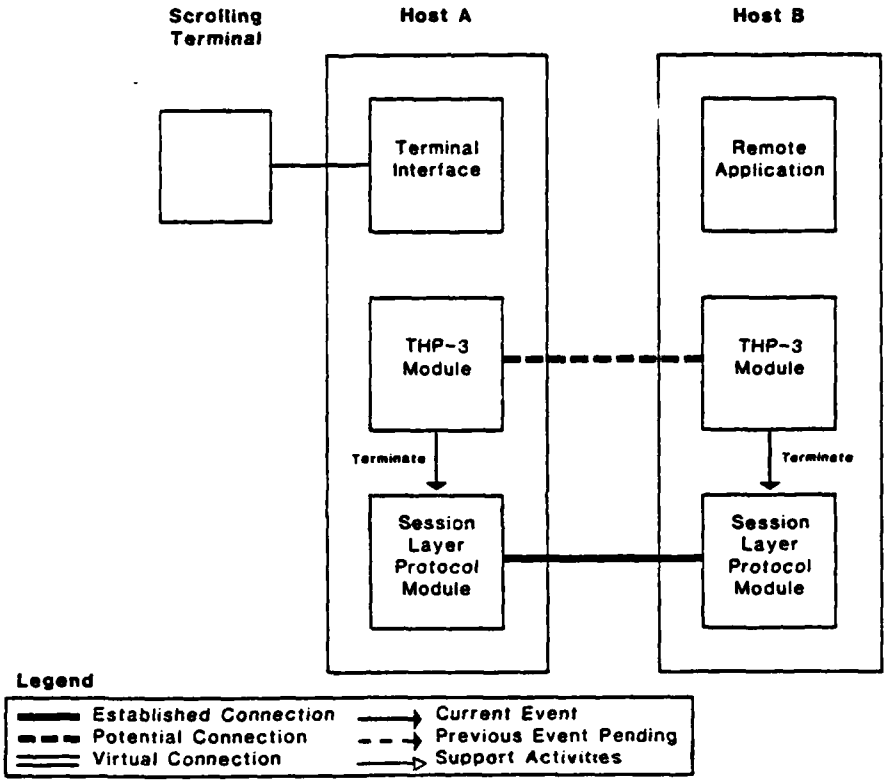


Figure 10. Termination of Support Services

d. the partner is able to control or interpret the usage of Devices of the Virtual Terminal.

2.2 DESCRIPTION OF MODEL COMPONENTS

2.2.1 Data Structure Definition

The DSD primarily defines how the cells of the CDS are organized for access, addressing, and control. Information stored in the DSD describes the structure and addressability of the information in the CDS. There are three distinct aspects to be described:

a. the type or types of atomic objects that the CDS is assumed to hold and the precise information that will be contained in each type.

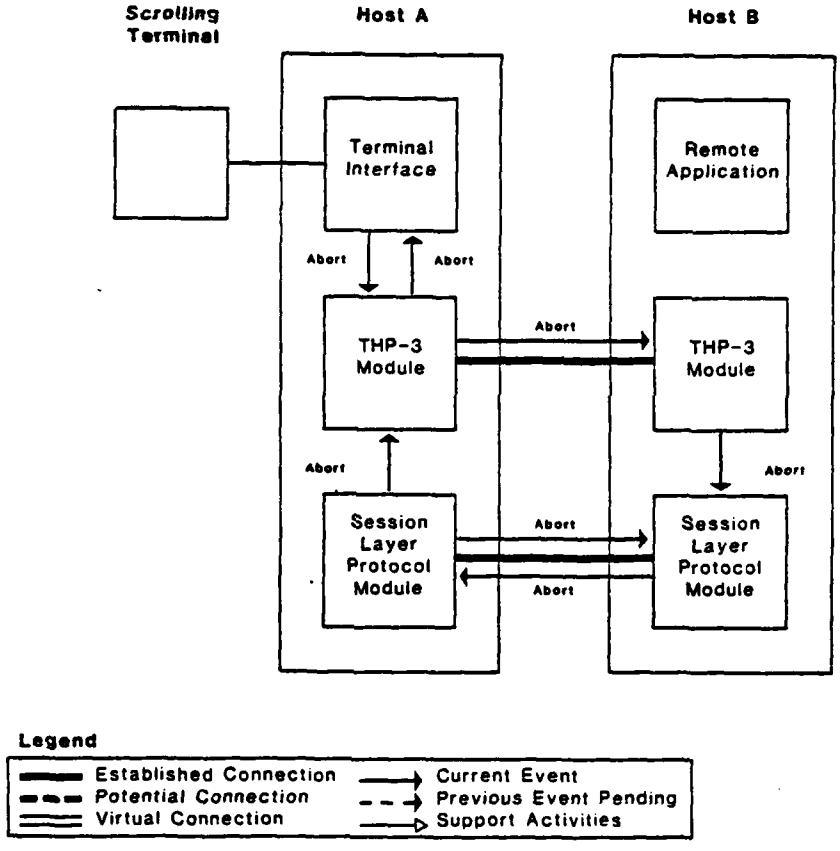


Figure 11. Origins of Abnormal Termination Requests

Examples:

Atomic Object: character; attributes: font, colours, highlight.

Atomic Object: vector; attributes: lines type, position references.

b. The structuring of atomic objects into composite objects and the attribute information which can be associated with such composite objects.

Examples:

Composite Object: Form composed of fields composed of several characters; attributes: protection, data validation.

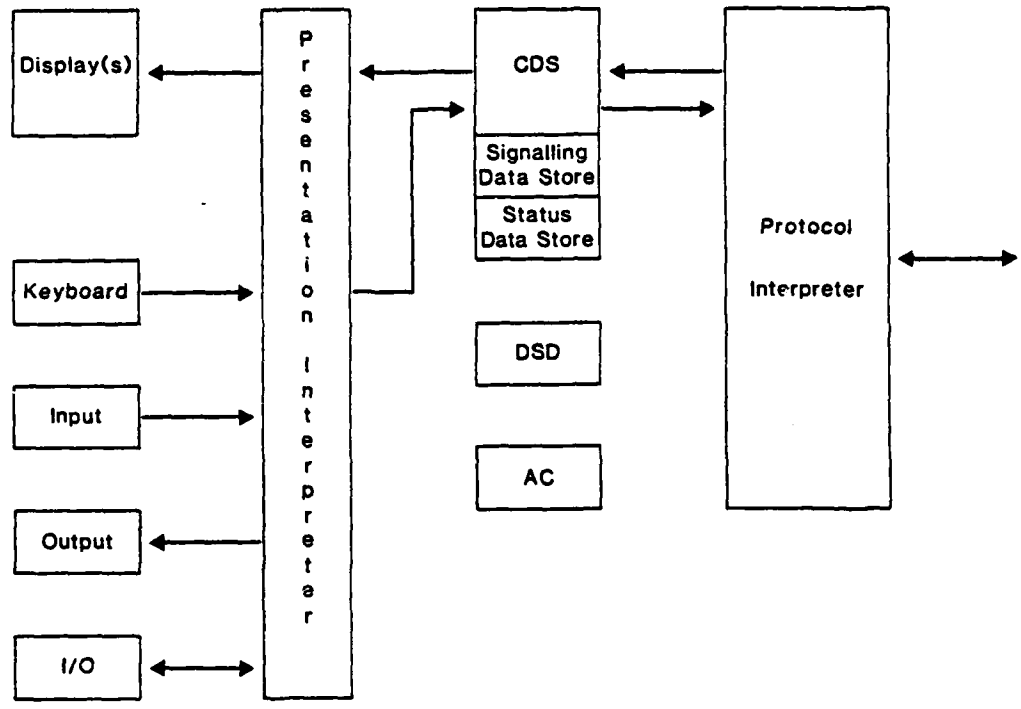


Figure 12. Virtual Terminal Model

Composite Object: Picture composed of several subpictures...composed of several graphic atomic objects; attributes: visibility, detectability, transformation...

c. The addressing mechanism(s) for addressing the atomic objects and any composite objects structured from them.

Examples:

1. Array of characters with XYZ addressing, whether each dimension of this array is bounded or unbounded.

2. Addressing of field composite objects by field name and/or forms (composed of fields) by form name.

3. Picture/subpicture/segment addressing for graphics

4. Chapter/paragraph addressing by sequence number for text.

The two partners are allowed by the Virtual Terminal Service to create a new description of the CDS and to delete the old description. A library of

descriptors may be created and managed for certain classes of terminals. The access of the DSD is controlled by the Virtual Terminal Service.

In addition the DSD also holds the parameters negotiated and accepted between the communicating partners; these are (for example):

1. class of service currently available
2. global parameters for interpretation relating to the object definition (allows the correct interpretation of the content of the cells of the CDS)
3. type of access allowed (two way simultaneous, two way alternate, etc.), and currently in force
4. other class dependent parameters, e.g.:
   - i. pointer to current Cell and/or Form or other complex structure being addressed
   - ii. current attribute(s) in use and defaults
   - iii. object attributes allowed, permissible values of each and default values

The DSD therefore holds all the necessary global information or profile of the Virtual Terminal Service as it is currently being used between the communicating partners. The exact way in which the DSD defines the CDS is class dependent.

2.2.2 Conceptual Data Store

The CDS provides the storage cells for Atomic Objects. The CDS consists of a finite or (potentially) infinite set of storage cells each capable of holding one Atomic Object. The objects of the CDS are typed. The structure and accessibility of objects are defined by the Data Structure Description. The storage cells are independent of each other except where a relationship is defined in the DSD.

An Atomic Object must have at least one component, but may have additional components:

Primary Value: this is the essential data content of the cell

Object Attribute Value: this part of the atomic object holds the particular values for that object of each Object in use.

The CDS is the abstract representation of the virtual terminal service's input and output. The screen of a real terminal can be considered a "window" to all or part of the CDS. The CDS can represent one or more input and output devices associated with a specific terminal. Inputs to the virtual terminal service manipulate the CDS. The virtual terminal service can control when the

peer CDS is updated by the transmission control facilities (see Section 3.1.6.2) and by the session quarantine services (see ISO [1981a, 1981c]). The virtual terminal service can also control when updates to the CDS are made visible to the application-entity by dialog control (see Section 3.1.7) or by presentation quarantine service.

2.2.3 Access Control

The Access Control provides the information whereby the access by each partner to the contents of the other parts of the CCA can be controlled. The complexity of this control, defined for a class, and negotiated for a particular presentation-connection, can vary widely. Access Control applies, in general, to the right to examine as well as the right to change, and in appropriate cases these can be separately controlled.

The Access Control handles access to the components of the virtual terminal for services such as dialog control and echo control. It also provides access control on individual elements of the CDS. The only access controls required for current purposes are dialog control, echo control, and protected fields. More sophisticated access controls require further study.

2.2.4 Signal Data Store

The SIDS holds signalling information being exchanged between the partners. This may be regarded as destined for the signalling device of the virtual terminal. The agreed signalling mechanisms (signalling device characteristics) are recorded in the DSD. The SIDS allows small pieces of information to be exchanged between the partners. This information is not necessarily subject to the same access control as the information in the CDS, which is why SIDS is shown separately. Signalling information would normally be presented to the underlying Session Service for expedited data flow transmission.

Every Signalling Mechanism defined will have an Identifier recorded in the DSD and a Value in the SIDS. These SIDS values can be accessed and manipulated in a flexible manner as defined in detail in the standards for the different classes. In general, however, the physical form of a Signalling Mechanism will not be specified in any class standard (or this document) as they are real device dependent. According to the requirements in a particular case, a particular physical feature may sometimes be used as part of the CDS and at other times be used as a Signalling Mechanism.

An example of a Signalling Mechanism is an Alarm Signal that can be activated at any time. Other examples could be

a. Attention keys

b. Tracking cursor

c. Bell

d. Light pen

A particular Class Standard may choose to define explicitly certain Signalling Mechanisms if these are an essential part of its operation. The provision by an implementation of additional Signalling Mechanisms not explicitly defined in the standard will generally be allowed.

2.2.5 Status Data Store

The SADS contains status information that can be exchanged between the two partners. There are various sources of status information, such as:

a. the status of a partner, or one of the subelements of the partner (e.g., one of the real devices used for the realisation of the virtual terminal),

b. exceptional conditions in the usage of the service,

c. reason codes, rejection causes, where transmitted to the other partner.

In general status information is only generated in case of abnormal conditions and is then put in the SADS to be available to the other partner.

The need to be able to prompt the partner to deliver certain information to the SADS is also recognised.

The status information is not necessarily subject to the same access control as the information in the CDS, which is why SADS is shown separately identified.

2.2.6 *Presentation Interpreter*

The Presentation Interpreter performs the mapping from the canonical representation of the data and operations to the representation required by the real terminal devices.

2.2.7 Protocol Interpreter

The Protocol Interpreter handles those aspects of the protocol not directly related to the virtual terminal itself. The Protocol Interpreter could be common to all presentation services. The Protocol Interpreter handles establishment, negotiation, data transfer, and termination services. It is responsible for interfacing with the session layer and for negotiating the appropriate session services. It may also be responsible for transmission control, e.g., character-at-a-time or line-at-a-time. More sophisticated transmission control functions, similar to the Remote Control Transmission and Echo option in Telnet [ARPA, 1973], may also be used.

2.3 SERVICE VIEW OF THE MODEL

The service view of the model consists of two virtual terminals connected to each other (see Figure 13). A virtual terminal is an abstract model of a terminal. It provides a model for describing the behavior of a real terminal without reference to the particular characteristics of that real terminal.

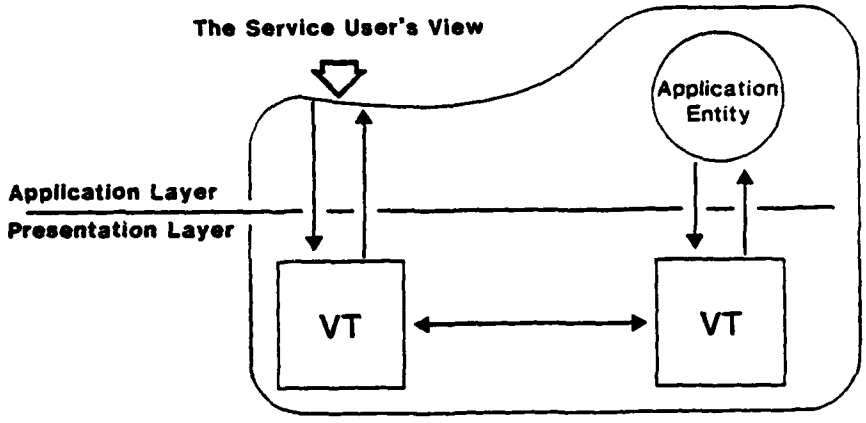


Figure 13. Service View of Virtual Terminal Model

Each virtual terminal is associated with an application-entity in that system. The view that the virtual terminal presents to the application-entity represents the current state of the virtual terminal and of the protocol exchange as observed by this side of the dialog. The states of the two virtual terminals will not necessarily be the same at any point in time, although this difference may not be observable to the application-entities.

Each application-entity makes service requests to the presentation layer to manipulate its virtual terminal. The results of these requests are sent to the peer virtual terminal and made available to the corresponding application-entity.

2.4 VIRTUAL TERMINAL CLASSES

The components of the virtual terminal model can be defined, structured, and accessed in different ways according to the services required. A single terminal protocol covering the range of possibilities would be extremely unwieldy and difficult to comprehend. On the other hand, separate protocols for each type of real device and mode of usage would require a very large number of implementations.

The concept of terminal class is introduced to make this situation manageable. Terminal classes organize the range of possibilities so that practical implementations for current terminal types and usages are possible. Classes also provide a framework for the development of more sophisticated features as they are required.

This report considers four classes of virtual terminal service:

a. Basic Class

b. Scroll Class

c. Paged Class

d. Forms Class

The Basic Class provides the most rudimentary service possible. In this class, it is possible to establish and terminate presentation-connections, negotiate new classes of service, and send data and signals. No canonicalization of data or terminal operations is provided.

The Scroll Class provides support for the simple line-oriented scrolling terminals, such as a Model 33 Teletype or a scrolling CRT. The class provides the Basic Class of service as well as canonical representation of characters and line-oriented operations, such as carriage return and new line.

The Paged Class provides support for page-oriented (i.e., two dimensional) terminals, such as CRTs, that use cursor positioning functions. This class provides the services of the Scroll Class as well as canonical operations for manipulating pages and their contents.

The Forms Class provides support for data-entry-type terminals. This class provides the Paged Class services as well as the ability to define forms with protected fields.

3. SERVICES PROVIDED TO THE UPPER LAYER

3.1 GENERIC VIRTUAL TERMINAL SERVICES

3.1.1 Introduction

The virtual terminal service is a service of the presentation layer. Once a presentation connection is established, all communication between peer entities is performed according to the rules for the class of virtual terminal service negotiated. The range of capabilities within a particular class is selected by further negotiation.

The remainder of this section describes the general features of the virtual terminal service. The service is partitioned into phases to facilitate the description and specification of the features. This partitioning in no way implies a particular scheme of implementation.

3.1.2 Overview of the Phases of Operation

Phases are well-defined aspects of a service associated with the occurrence of prescribed events or conditions. The virtual terminal service consists of five phases (see Figure 14):

Null

Establishment

Negotiation

Data Transfer Service

Termination

The virtual terminal service is connection oriented. Use of the service is therefore preceded by a connection establishment dialog and followed by a connection termination dialog. The virtual terminal service also provides a negotiation facility for selecting the appropriate context of interaction for the large variety of possible data structures. All presentation layer services require establishment, negotiation, and termination. These may eventually become standardized presentation services, but for the present no such generic service is assumed.

Null Phase. Any presentation-entity is in the null phase before the request for presentation services is made.

Establishment Phase. Establishment is the initial phase of service. The establishment phase is the only service phase that may be entered before establishing a presentation connection. A request for virtual terminal service initiates a sequence of activities among presentation-entities to establish a presentation connection.

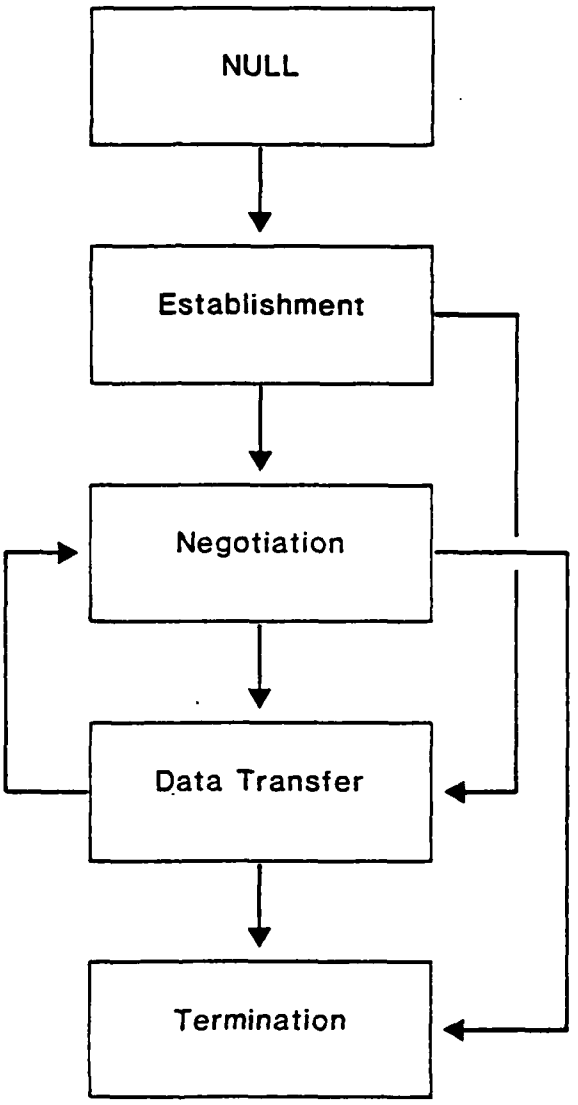


Figure 14. Virtual Terminal Service Phases

Establishment services may either succeed in establishing a connection or fail due to circumstances in the lower layers or refusal by the remote presentation layer. If unsuccessful, the presentation-entity notifies the upper layer and returns to the null phase. If successful, a positive response is returned and either the negotiation phase or the data transfer phase is entered.

Negotiation Phase. The negotiation phase of service may be entered from the establishment phase or the data transfer phase to modify parameter values. Most parameters of the virtual terminal service are class dependent. Negotiable parameters may govern data structuring and access schemes. They may also control the availability of features and the schemes for their use.

Negotiations may fail to establish agreement on the diverse range of contextual parameters avqilable. Various compromises allow resumption of the data transfer phase with unchanged or partially changed contexts, but if agreement cannot be reached, the application-entity may choose to terminate the service.

Data Transfer Phase. During the data transfer phase, the presentation-entities can exchange data, signals, status, and commands. The data exchanged consists of characters, lines, pages of text, form labels and field values, or graphic data, depending on the current class of service. Signals necessary to support the orderly exchange of data and commands and to communicate application exceptions are class dependent. Status information also varies with the class of service and negotiated terminal capabilities. Both normal and expedited transfers will generally be available.

Commands for manipulating data attributes, parameters, or modes of activity are class dependent except for those relating to service phase transitions. At any time in the data transfer phase, an application-entity may initiate entry into the negotiation phase. The data transfer phase may be exited only for negotiation or for termination of service.

Termination. The termination phase can be entered from the data transfer phase or the negotiation phase. It provides for the orderly release of the presentation connection and termination of virtual terminal service. The termination phase also allows a presentation connection to be aborted with possible loss of data. The termination phase is never entered from the establishment phase since the latter performs all the necessary recovery activities if a presentation connection cannot be established.

3.1.3 Establishment

Establishment is concerned with selecting the presentation service required by the service user initiating the presentation-connection, agreement of this by the other partner, and, where necessary, negotiation of other parameters of presentation service. In its fundamental aspects, establishment is common to all types of presentation service since it must be capable of selecting any available service from an initial state in which there is no presentation service at all.

It will normally also be concerned with establishing a session-connection with the appropriate characteristics to support this presentation layer activity;

some aspects of the session service may be of direct interest to the service user rather than to the presentation layer functions as such.

It is possible to modify session service characteristics during a particular presentation-connection. The establishment phase is concerned with performing the synchronization procedures between peer entities to establish the presentation-connection. During this phase the initiating entity may also specify the particular presentation service and class of service desired. It may also be possible to perform initial negotiation by specifying a predetermined profile.

3.1.4 Negotiation

Although included in this document, this Service is seen as being essentially independent of the type of presentation service being used. Thus for example it may be applicable to the selection/negotiation of the type of service required by the potential partners attempting a presentation-connection (in cases where this is not known in some way from their identities).

The service described in the following material is conceptually applicable to each and every separate parameter independently. This is potentially very complex and in practical cases groups of parameters may be negotiated together. However, there is likely to be, in all but the most simple of cases, a hierarchy of parameters, i.e., some cannot be determined until others higher in the hierarchy are known (class is an example of a high-up parameter), and thus the negotiation service (and protocol) will progress in a number of stages. It is quite possible that several such stages will be in progress at the same time. However, where appropriate, the service can be subset down to a very simple operation.

The negotiation service consists of an optional information subphase and the selection subphase. During the information subphase, an entity may either inform its peer entity of the parameter or attribute values it supports or request the peer entity to inform it of the values supported by its peer. During the selection subphase, the peer entities negotiate specific values for the parameters.

Negotiation is terminated when agreement is reached or when the negotiation is rejected. If no agreement is possible, the parameter values are set to the values before the negotiation phase began. The user may now either terminate the virtual terminal service, start a new set of negotiations, or continue the data transfer phase.

3.1.4.1 Default Values

For each parameter a default value, or a default range of values, shall be specified as part of the parameter definition. A possible default is "non-existing."

At connection establishment time, before initial negotiation, all parameters are set to their default value(s).

A parameter that is not explicitly negotiated during the negotiation phase shall keep the value or range of values it had before the negotiation phase (it will be its default value if it is the initial negotiation).

3.1.4.2 Profiles

A profile is a complete set of parameter values or value ranges that is standardized and known by all potential partners. A profile is designated by a profile identifier, with certain mandatory parameters (in the general case).

It is negotiated as a whole; after an agreement on a profile has been reached, some of the parameters of the profile can be individually renegotiated.

3.1.4.3 Dynamic Parameter Switching

The selection of several values for a parameter means that dynamic change of the value of this parameter can occur during the data transfer phase, among the selected values only.

3.1.4.4 Quality of Mapping

An indication associated with some parameters can give some information on the required quality of mapping of these parameters to real characteristics of the partner.

3.1.5 Termination

The termination service provides the means to terminate an instance of presentation service. Termination may be of two forms: the normal "finish" request and the "abort" request. Termination is common to all types of presentation services. The finish request terminates the presentation connection without loss of data. The abort request is an immediate termination, and some data may be lost.

3.1.6 Data Transfer

This phase is also seen currently as specific to Virtual Terminal Service although an analogous phase will probably need to exist in other presentation services. The precise mechanisms may be different. The following material gives some general aspects of this part of the Virtual Terminal Service. Each individual class standard will need to build on this general information with the detail applicable to its methods of operation. Those aspects that are defined herein in some detail are expected to be applicable to all or at least most classes but are likely to be repeated in the class standard to make this complete and so that the conformance statements of that standard can be made to apply. A class standard should indicate if any parts of this general information are not applicable to it.

3.1.6.1 Entry of Information to the CDS

This service is concerned with the entry by one partner of information into the CDS for subsequent visualisation of the resulting content of the CDS by

the other partner as the conceptual image.

The next storage location at which data is to be entered is marked by a pointer. The location so indicated is known as an Active Location. The capabilities for explicit moves of an active location are determined by the facilities of the class and the negotiated parameters of the CDS. The possibility of multiple pointers/active locations is not excluded.

The Service Interface primitives that support this service are class dependent. It is expected that each class will provide one or more primitives of the following general types:

TEXT object-primary-value
(nature of object-primary-value is class dependent)

- enables the value of the Primary Value of the Object at the/an active location to be adjusted.

[POSITION-CONTROL]
(class dependent names and parameters)

- enables, according to the capability defined for a class, movement of the/an active location (within whatever bounds or limits are placed on such movements by the definition of the CDS, the parameters negotiated for it, and the movement capabilities declared at negotiation time).

In a multi-pointer case a means of selecting or designating the intended pointer will be needed.

The movement of the active location may be independent of access control on the CDS, but the entry of data (including change of attributes) is only possible if the partner has Write Access to the specific part of the CDS currently selected by the pointer. Free movement of the pointer may be considered desirable so that all parts of the CDS can be reached conveniently, but this is implementation dependent.

3.1.6.2 Transfer Control Services

The control of data transfer may be exercised by three different services:

a. Session quarantine

b. Transmission control

c. Presentation quarantine

The session quarantine service controls the updating of the peer CDS by controlling when data is released by the session layer. Transmission control service controls the updating of the peer CDS by controlling when data is given to the session layer. Transmission control can be used in conjunction with session quarantine. Presentation quarantine controls when information in

the CDS is released to the application-entities.

3.1.6.3 Access Control

Three levels of access control service are provided by the virtual terminal service:

a. Access control on the virtual terminal service as a whole, i.e., strict dialog control

b. Access control on the local or remote CDS as a whole, i.e., echo control

c. Access control on the atomic elements contained in the virtual terminal

Dialog control controls application-entity write access to the virtual terminal service. Dialog control is exercised at the boundary between the application and presentation layers. Dialog control does not affect the access of the CDS by a peer entity. Such access control is exerted by a combination of the transmission control services and the other access control services.

Echo control controls application-entity write access to the local or remote CDS. Input from an application-entity is sent directly to the remote CDS or is first entered in the local CDS and then transmitted to the remote CDS. Echo control may be exercised by either peer and may be used in conjunction with transmission control services to allow more sophisticated services.

It is also possible to control access to the atomic elements of the virtual terminal service. This access control coordinates and controls the access within the CDS as well as the values of various attributes. It could be used to provide protected fields in a forms class. The details of this access control service require further study.

3.1.6.4 Attributes

Control of the Attributes information in the CCA is an important part of the operation of a Virtual Terminal Service. As indicated earlier, Attributes fall into two major categories:

a. those which are, at least conceptually, contained within the individual Cells of the CDS so that they form part of the Atomic Object in each Cell and qualify the Primary Values of these Objects (the fact that some or all such attributes may, in a particular case, be able to be held on a once-per-CCA basis does not alter this classification or the way in which their use is defined); this category of attribute is known as Object Attributes,

b. those which are contained outside the CDS because either they are inherently applicable to the whole CDS (for example those which define grouping of cells) or are related to other components of the CCA; this category is known as General Attributes and subclassifications are likely to arise when the full variety has been studied further.

In both categories attributes are only manipulated at negotiation time and are fixed during the normal operation phase. For others primitives will be provided to allow the values of the attributes, and possibly the scope of applicability, to be changed explicitly. Some attributes may also be affected implicitly by other primitives. These two characteristics of attributes, orthogonal to the above classification, are known as Constant and Variable, and can qualify the above classification.

Ability to alter Attribute Values is subject to Access Control; this may or may not be related closely in particular cases to the access control on the CDS cells. For example, in Basic Class, manipulation of the Attributes Values of a cell or cells of the CDS is always conditional on having Write Access to the whole CDS, this being the form of access control defined in that class. In a more complex class, particular sets of attributes, possibly with a hierarchic structure, may be defined as having an access control token associated with them, so that the right to alter them may be controlled independently of other elements of the CCA. It is also important not to confuse the right to alter the value of a negotiated attribute within some negotiated range with the right to re-negotiate the range itself.

Character Attributes. For Character Oriented Classes, the CDS can have associated with it one or more Character Attributes. This implies that each Cell has the capability to store, at least conceptually, a Value for each of these attributes. The range of values permissible, and whether the value may vary across the CDS is negotiated for each parameter and the result of the negotiation recorded in the DSD.

In the following material, each attribute is defined in terms of two aspects:

a. meaning of the attribute

b. set of permissible values

Permissible values for a particular presentation connection are negotiable. For all attributes a default value is defined, indicated by underlining one of the values. If the use of this attribute is negotiated then it will take the default value unless negotiation establishes a different value.

For the purpose of erasure two types of attributes are recognized:

a. Attributes, the value of which is always reset to the default or initial value if the cell is erased; These are called the linked attributes.

b. Attributes, the value of which is only reset to the default or initial value during erasure, if such resetting is indicated via a parameter. Otherwise such attribute does not change during erasure; These are called the non-linked attributes.

The following list covers the cell attributes currently defined for character-oriented classes. This list may be added to from time to time as new capabilities arise.

Character Encoding

Meaning: specifies the code used for the representation of the characters of the character repertoire in the CDS cells.

Permissible Values: ASC11-7, ASC11-8, encoding of ISO set for text communication (to be confirmed), user defined values.

Character Repertoire

Meaning: specifies the set of characters which can be written in a CDS cell. Character sets can be standard ISO registered sets or user defined.

Permissible Values: IRV, US, ISO character set for text communication (to be confirmed), user defined set ids.

Note 1: subsets of registered character repertoires can be defined, such as the "upper case" subset. They are given a specific identification.

Note 2: specification of a protocol for transferring definitions of user defined character sets across an Open System Interconnection is not part of this document. User defined character sets can still be used successfully but until such time as a protocol for transferring their definitions is standardized it is assumed that the definition is entered at both ends of the OSI connection across which it will be used (according to the local standardized definition facilities which may be completely different). Agreement to use a particular user defined set is negotiated and will only be successful if the user-defined-character-set-id is recognized at both ends. Ensuring consistency of local definition is a user problem completely outside the concern of this document.

Controls Encoding

Meaning: specifies the method used for control information representation.

Permissible Values: VT controls, embedded (ECMA-48).

Note: As experience with Telnet and THP have shown, it is much more efficient if controls and data are encapsulated by protocol headers. This avoids scanning every character looking for a control that may need to be transformed for output to the real terminal. However, the ECMA-48, or extended ASCII, code is also supported for those cases where translation can be avoided completely.

Character Font

Meaning: specifies the Font in which the Characters will be presented.

Permissible Values: as available, italics, pica, elite, user defined fonts ids.

Colour Capability

Meaning: specifies the kind of colour rendition that is required for the presentation of data.

Permissible Values: black and white, discrete colour for characters, discrete colour for characters and background.

Note: More general colour capability is for further study.

Colour Set

Meaning: determines the set of colours available for character presentation, when there is colour capability.

Permissible Values: ECMA-48 set of colours: WHITE, RED, YELLOW, GREEN, BLUE, MAGENTA, CYAN, BLACK.

Note 1: variability on a character-by-character basis is expected to be provided if more than one value is negotiated for this attribute.

Note 2: More general colour capability is for further study.

Character Intensity

Meaning: determines the intensity at which the Character is presented.

Permissible Values: 0, 1, 2, 3, 4, 5, 6, 7. 0 is equivalent to Concealed or Invisible.

Note 1: variability on a character-by-character basis is expected to be provided if more than one value is negotiated for this attribute.

Note 2: the effect that variable intensity has on a coloured character on a coloured background is a characteristic of the real presentation device. At zero intensity the background is expected to be retained.

Inverse Rendition

Meaning: if monochrome, light and dark are interchanged (over the extent of the object). if coloured, character and background colours are reversed.

Permissible Values: normal, inverse

Note: variability on a character-by-character basis is expected to be provided if more than one value is negotiated for this attribute.

Flash Character

Meaning: intensity of Character varies periodically from zero to value determined by Character Intensity attribute.

Permissible Values: Normal, Flash (provision for variable flash rates is for further study).

Note: variability on a character-by-character basis is expected to be provided if more than one value is negotiated for this attribute.

Level of Emphasis

Meaning: some method is required/available for varying the emphasis of the presentation of the information but the actual method to be used is not specified (thus giving maximum freedom to the local mapping functions).

Permissible Values: 0, 1, 2, 3, 4, 5, 6, 7. '4' is deemed to be 'normal,' '0' to be 'not perceptible.'

Conceal

Meaning: If concealed, the character is displayed with the same display attributes as the background field. The value concealed disables the following attributes as follows:

colour: the character colour is that of the background

intensity: the character intensity is that of the background

inverse: the whole field is displayed inverted

flash: no flashing of the field

underline: an underline is not displayed

Permissible Values: normal, concealed

Note: variability on a character-by-character basis is expected to be provided if more than one value is negotiated for this attribute.

Underline

Meaning: when underlined, the character is displayed with a horizontal line below the character.

Permissible Values: normal, underlined

Note: variability on a character-by-character basis is expected to be provided if more than one value is negotiated for this attribute.

3.1.6.5 Data Structuring

The virtual terminal service provides several means for structuring data in the CDS. One major category of this structure is for character-oriented terminals.

Character-Oriented Data Structuring. The semantically indivisible (atomic) objects for Character-Oriented Classes are single characters. These will often be the well known sets of alpha-numeric and punctuation and other symbols, but the use of mosaic characters, or of user defined symbols, or of purely abstract entities with no visual representation is possible. Facilities are provided for selecting the set of characters applicable, how these are to be encoded, and how these are to be represented in the Conceptual Image when this is semantically significant.

The structuring information in the DSD describes how the cells are arranged into an ordered array of one, two or three dimensions each of which may be bounded or unbounded. This provides great flexibility in local mapping to real devices. Such structuring information is placed in the DSD during negotiation for a presentation service connection. Renegotiation of the class changes the structuring of information.

Each Atomic Object is defined to have two components:

Primary Value: this represents a single "character,"

Attribute Values: the Cell Attributes are Character Attributes, i.e. apply to a single character.

The Values given to the various Character Attributes determine the meaning of the Primary Value and can also give various secondary characteristics to the object, for example determine the style of visual imaging of the object on a screen or printer.

For each such attribute there will be a valid range of permissible values and a default value which will apply initially to all Cells of the CDS. Further definition of attributes is given in Section [3.1.8].

Data Structure Definition. The DSD holds the parameters which define all aspects of the CDS (including its existence), including the way in which the cells of the CDS are organised for the purposes of access, addressing and control. It also holds the parameters of the atomic object definition, in particular the character attributes which are to be applicable and for each one, the range of values, the default value, and whether it can be variable across the CDS.

The organisation of the Cells is defined in terms of a number of "Dimensions." These are independent of each other and each can be bounded, i.e. finite, or unbounded, i.e. potentially infinite, and each can have certain position control capabilities. The dimensions define the ordering of the Cells and hence the only interrelationship between the contained Objects explicitly. A bound to a dimension defines in respect of that dimension the set of coordinate

values which are valid as part of the designation of a Cell. The Pointer consists of 1, 2 or 3 such (valid) coordinates and thus designates (addresses) the one Cell at the active location. An unbounded dimension implies that there are no valid values 'before' its Origin but that an indefinite set of values is available in the other direction; individual values are assigned, i.e. 'come into existence,' when the active location is moved to them and the cells then remain in existence (they may not remain addressable within the VT - this depends on the position control capabilities). The possibilities provided for parameterisation of the dimensions allow a variety of device capabilities and usage requirements to be catered for.

X Dimension

This dimension represents a one-dimensional ordered array of storage cells. X is considered to be horizontal where this is meaningful, and an X-array is commonly known as a Line, composed of Characters.

X = 1 is the Origin of X and is the start of the X-array and designates the location of the first cell.

The X-array can be unbounded or bounded. If bounded, a bound (or maximum X coordinate) determines the number of storage cells (or size) of an X-array. If unbounded, an X-array is of variable size. Its current size is determined by the maximum value of X when a cell is written to. New cells will be generated as required by such writing in the X dimension. If Y is defined each X-array can have a different current size. Once generated the cells continue to exist although they may not be addressable within the position control capabilities of the VT.

The following methods of movement on the dimension are defined and are subject to negotiation except as stated below to be 'always available' or 'not provided':

Sequential implicit: caused by entry of text into the active location. Always available but see below for an explanation of limit if X is unbounded.

Differential: move forward or backwards (X increasing or decreasing) relative to the current active location and within a declared range of movement; also the destination of the move must be within the declared range of X. Backwards moves may not be possible (declared capability at negotiation time).

Absolute: direct addressing (from current X to higher or lower value) within the bound. Move to lower value may not be possible (declared capability at negotiation time).

Home: set the pointer to address the origin of X. May not be available (declared capability at negotiation time).

For the X dimension there is defined a limit which is applicable only when X is bounded and when the pointer has an X coordinate of value (X-bound + 1).

This is reached by the implicit move of the pointer caused by writing into the last cell of the X-array. Further entry of text when the pointer has this value is not valid.

Movement on the X-dimension never causes implicit move on any other dimension.

Y Dimension

This dimension, when defined, represents in conjunction with X a two-dimensional array of storage cells. Y is considered to be vertical where this is meaningful, and an X-Y-array is commonly known as a Page, composed of Lines of Characters. If not defined only one unique X-array can ever exist in the CDS.

Y = 1 is the Origin of Y and designated the location of the first X-array of an X-Y-array.

An X-Y-array can be bounded in the Y dimension. If bounded a bound (or maximum Y coordinate) determines the number of X-arrays (or depth) of a Y-array. If unbounded, a Y-array is of variable size. Its current size is determined by the maximum value of Y when a cell on that Y-axis is written to. New X-arrays, bounded or unbounded, will be generated by such writing in the Y dimension. If Z is defined each Y-axis may have a different current size. Once generated the X-arrays continue to exist even though they may not be addressable within the position control capabilities of the VT.

The following methods of movement on the dimension are defined and are subject to negotiation except as stated below to be 'always available' or 'not provided':

Differential: move forward or backwards (Y increasing or decreasing) relative to the current active location and within a declared range of movement; also the destination of the move must be within the declared range of Y. Backwards moves may not be possible (declared capability at negotiation time).

Absolute: direct addressing (from current Y to higher or lower value) within the bound. Move to lower value may not be available (declared capability at negotiation time).

Home: set the pointer to address the origin of Y. May not be available (declared capability at negotiation time).

Note: Movement in the Y-dimension never causes implicit move on any other dimension.

Z Dimension

This dimension is not mandatory; it can be defined when both X and Y are defined and in conjunction with X and Y represents an ordered set of X-Y-arrays. If not defined only one unique X-Y-array can ever exist in the CDS. A X-Y-Z-array may where meaningful be considered as a group or chapter of

pages.

Z = 1 is the Origin of Z and designates the location of the first X-Y-array (page).

The X-Y-Z-array can be unbounded or bounded in the Z dimension. If bounded, a bound (or maximum Z coordinate) determines the number of X-Y-arrays (or thickness) of an X-Y-Z-array. If unbounded, the number of X-Y-arrays is of variable size. Its current size is determined by the maximum value of Z when a cell in the X-Y-array is written to. New X-Y-arrays, bounded or unbounded in each dimension, will be generated by such writing in the Z dimension. Once generated such arrays continue to exist even though they may not be addressable within the position control capabilities of the VT.

The following methods of movement on the dimension are defined and are subject to negotiation except as stated below to be 'always available' or 'not provided':

Differential: move forward or backwards (Z increasing or decreasing) relative to the current active location and within a declared range of movement; also the destination of the move must be within the declared range of Z. Backwards moves may not be possible (declared capability at negotiation time).

Absolute: direct addressing (from current Z to higher or lower value) within bound. Move to lower value may not be available (declared capability at negotiation time).

Home: set the pointer to address the origin of Z. May not be available (declared capability at negotiation time).

Non-Character Oriented Data Structuring.

TO BE SUPPLIED

3.1.6.6 Signalling

Signalling services are provided via SIDS and DSD (independent of the CDS and the access control on it) to allow definition, use, and interrogation of such features.

The nature of signalling mechanisms is not defined explicitly in this document. A particular Class may choose to define explicitly certain Signalling Mechanisms if these are an essential part of its operation.

Two forms of signals are provided by the virtual terminal service:

a. Asynchronous signals

b. Synchronous signals

The asynchronous signal is sent as an expedited-service-data-unit. A response is not required. Transmission of an expedited-service-data-unit is not directly related to the normal flow of data. An expedited-service-data-unit is guaranteed to be delivered before any subsequent data-unit on this connection.

The synchronous signal sends an expedited-service-data-unit that is also coordinated with the normal data flow. The coordination is provided by the use of a data mark in the normal data stream. A response is required. The general token service of the session layer may be used to support this service.

Failure to deliver data to a signalling mechanism due to inoperability will cause an error to be reported.

Use of an unavailable sig-mech-identifier, i.e. one not in the negotiated set, is invalid.

Operations on signalling mechanisms will be handled using an Expedited Data Session service (and not subject to any presentation access or transmission control); in any event delivery is guaranteed to be no later than the effect of any subsequent Presentation service primitive.

3.1.6.7 Auxiliary Devices

The information content of the CDS (the conceptual image) can be displayed on or written into by one or more devices, possibly subject to the control of such devices by the other partner, and subject to the access control on the CDS.

The nature of the different devices is not defined explicitly in this document. A particular Class may choose to define more than one Device.

Device control on devices is maintained in sequence with the effects of other normal presentation service primitives. Some classes may require selective access control of CDS information by Virtual Devices. Independent quarantining for different Virtual Devices is for further study.

3.1.6.8 Status Services

The status of all components of the Virtual Terminal are held in the SADS by the presentation entity and can be interrogated by the other partner or will be sent spontaneously when some change of status occurs. Access control does not apply to the status device.

3.1.6.9 Error Reporting

A number of error situations are identified as below and the actions taken are as indicated below. (This material is at an early stage of development.)

a. detection by a presentation entity of incorrect use of the service interface by the local presentation-user.

The service interface primitive is rejected and no action taken on the CCA. No protocol is generated. (Other user is not advised.)

b. detection by a presentation entity of incorrect protocol elements or sequences from the other presentation entity.

The presentation entity will generate a request for re-synchronization and transmit this to the other presentation entity; this action may require explicit presentation protocol but will generally use the session synchronization service. The local presentation user is not notified at this stage unless the extent of recovery makes rejection of one or more quarantine units necessary.

In (a) and (b) error conditions include violations of parameter values, whether fixed or negotiated, and as defined in class standards, and violation of access control.

c. detection by a presentation entity of a request to re-synchronize from the other presentation entity; this may be due to a protocol error by this presentation entity or to a failure in the lower level possibly not detected or notified explicitly by session layer.

The presentation entity will respond to the request by attempting to effect a resynchronization. This may require rejection of one or more quarantine units (PSDUs) at the local service interface.

The actions in (b) and (c) are still being studied.

d. attempts to re-synchronize are unsuccessful due to a failure of the presentation entities to achieve agreement (one presentation entity may have become corrupted).

e. session-connection appears unusable (effect may be detected by one presentation entity only).

f. session-connection is irretrievably lost. In these cases a presentation-connection-abort indication will be given to both presentation users (or at least the one local to the presentation entity which detects the condition) and the presentation layer returns to Phase 1 - no presentation-connection.

3.2 BASIC CLASS VIRTUAL TERMINAL SERVICES

3.2.1 Introduction

The Basic Class of the virtual terminal service supports the exchange of data in the form of character streams. These services constitute the minimal character oriented class of the virtual terminal service. No primitives facilitating canonical interpretation or manipulation of data are provided. No mechanism is provided for structuring the stream into lines or pages. Basic Class services provide limited support suitable either for simple terminal applications or for complex specialized applications requiring extensive

control over the data stream.

All character stream devices may be conceived of as basic terminals. Some examples include paper tape readers/punches, character printers, streamed news displays used by television and investment brokers, automation control keypads, and most communications terminals. Control codes for these devices are typically communicated on the same channel with normal data. The Basic Class is intended for use with these very simple or very complex devices for which the other classes of the virtual terminal service are inappropriate. However, applications for generating and interpreting the particular control codes for each device must be provided. As specific patterns of usage of Basic Class services become apparent, they may serve as a basis for standardization of further classes of the virtual terminal service.

3.2.2 Model Characteristics

The Conceptual Data Store contains a single storage cell for each basic entry device and each basic display device of the negotiated terminal configuration. These storage cells are called entry or display streams as appropriate and represent streams of character data for communication with remote entities. Each stream has an associated attribute indicating whether it is empty or contains active data. Entry of subsequent data into a stream is governed by the processes that realize the exchange of data with remote entities.

Application programs are typically designed to view an entry stream as a read-only structure and a display stream as a write-only structure. Terminal users typically adopt the opposite view that data is read from a display stream and written using the entry stream. To avoid this confusion and provide a conceptual framework for modelling Basic Class services independent of the nature of the application-entities involved, an alternative nomenclature is employed.

Character streams are also called data sources or sinks. A source presents a stream of characters transferred from a remote partner, and a sink can accept a stream of characters to be transferred to a remote partner. Thus, negotiation of a virtual terminal configuration leads to agreement between partners on the numbers of sources and sinks of each and their possible interconnections. Moreover, sources and sinks may be mapped into entry and display streams in any fashion required locally.

Each source and each sink has an identifier for use with device control services. The elements of the model for use with Basic Class services do not otherwise require parameterization. The concept of character data is undifferentiated. No services regarding character attributes or graphic rendition are provided. Characters are specified only as octets (8-bit binary codewords).

3.2.3 Minimal Services

Establishment. Establishment services are described in the discussion of the generic services.

Negotiation. Negotiation services are described in the discussion of the generic services.

Termination. Termination services are described in the discussion of the generic services.

Data Transfer. The following services are required for the data transfer services of the Basic Class. These services are described in the discussion of the generic services:

a. Normal transfer of character string data

b. Expedited transfer of asynchronous signal data

c. Expedited transfer of synchronous signal data coordinated with the normal data stream

d. Device control for multiple device terminals

e. Transfer control services

f. Access control services

g. Status report services

h. Error report services

3.2.4 Additional Services

No additional services are currently specified for the Basic Class of the virtual terminal service.

3.3 SCROLL CLASS VIRTUAL TERMINAL SERVICES

3.3.1 Introduction

Scroll Class services support the exchange of lines of data. Lines are sequences of character strings punctuated by control elements to indicate the intended interpretation of the data. A line is the nominal unit of interaction between application-entities employing Scroll Class services.

A variety of so-called scrolling terminals exist. The term "scrolling" is suggested by the frequently limited segment of the interaction dialog made visible to the user. The data stream is shifted through a viewing window and then disappears. However, neither the viewing window nor the association of entry and display devices for dialog presentation are essential characteristics of such terminals. The line oriented mode of interaction is the characteristic of scrolling terminals forming the basis for specification of Scroll Class services.

Scrolling terminal devices exhibit operating characteristics particularly suited for manipulation of data in line units. Support of such operation is

often distributed between the actual terminal and its system interface. Scrolling terminal display devices map data lines into lines of character graphics. Control elements may be used to select modes of graphic rendition or to format the display line. Scrolling terminal entry devices construct data lines from sequences of characters and control elements selected by the user. Control elements may be selectable during entry for delimiting keywords, for indicating the entry device used, or for requesting special processing of data.

Scroll Class services accommodate a range of diverse terminals. Most of these feature a limited set of control functions governing graphic rendition and other presentation parameters. Most video and hardcopy terminals may be utilized by means of Scroll Class services. Line printers may also be operated as write-only scrolling terminals. Many terminals with graphics capabilities feature character oriented capabilities usable by means of Scroll Class services.

Most scrolling terminals consist of a single entry device associated with a single display device. This configuration is the default for use with Scroll Class services. However, the virtual terminal service allows alternative device configurations to be negotiated.

The application-entities that interact by means of the virtual terminal service may be human users or application programs. The choice of control elements useful in the exchange of lines is constrained by the nature of the entities, the nature of the interaction, and the nature of the devices. Scroll Class services include control elements for display control, data format control, and transfer control.

3.3.2 Model Characteristics

The Conceptual Data Store contains a one dimensional array of storage cells for each scrolling display device and each scrolling entry device of the negotiated terminal configuration. These storage cells are called character positions, and the arrays are called display or entry lines as appropriate. A character position can hold a single character and the associated attributes governing its display and interpretation. Each character position also stores an attribute value to indicate the presence or absence of data in the position.

Character attributes, described in the discussion of generic services, may be partitioned into groups based on similarity of function. For the Scroll Class, character attributes are partitioned into character set attributes and graphic rendition attributes. The character set attributes determine a specific association between encoded data and symbolic interpretations, whereas the graphic rendition attributes affect the graphic images of characters but not their identities.

Parameters associated with a line include length, tab settings, and overflow handling. Overflow handling determines whether carriage-returns or new-lines are automatically inserted when long data lines overflow the line length. Other alternatives include truncation and entry disablement. A cursor

pointing to the next character position into which data will be entered is also maintained for each line. Echoing may be effected by duplication of entry line data in a display line.

3.3.3 Minimal Services

Establishment. Establishment services are described in the discussion of the generic services.

Negotiation. Negotiation services are described in the discussion of the generic services.

Termination. Termination services are described in the discussion of the generic services.

Data Transfer. The minimal servises provided by the Scroll Class include all minimal services of the Basic Class. Additionally, the following services are required for the data transfer services of the Scroll Class:

a. Explicit indication of completion of a line (clears the line to prepare for subsequent line transfers)

b. Sequential deletion of characters at the cursor position

c. Cancellation of an incomplete line to allow transfer of an alternate line

3.3.4 Additional Services

Data Transfer. The following services may be implemented to allow the Scroll Class to support the additional features of some scrolling terminals. Use of these services is subject to negotiation:

a. Selective erasure of portions of an incomplete line

b. Positioning of a cursor to the beginning of its line

c. Backspacing of a cursor within its line

d. Horizontal tabulation services

e. Dynamic selection of alternative character sets

f. Dynamic selection of graphic rendition parameters

3.4 PAGED CLASS VIRTUAL TERMINAL SERVICES

3.4.1 Introduction

Paged Class services support the exchange of paged data. Paged data may consist either of pages of data or of page fragments with controls indicating their relationship to pages. A page is a two dimensional arrangement of text

consisting of an array of lines. Paged data streams are identical in form to line oriented data streams. Both consist of sequences of character strings punctuated by control elements to indicate the intended interpretation of the data. However, additional types of control elements may occur because of the additional interpretations relevant for paged data.

Most page oriented terminals employ CRT display technologies. However, the mechanics of producing an image is not relevant for utilization of Paged Class services. The essential characteristic of page oriented terminal applications is that data is structured in page units. Such structuring may be provided in hardware or software. The capability for rapid modification of a previously displayed page is desirable for support of interactive applications.

Paged terminal devices exhibit features suited to manipulation of data pages including relative and/or absolute cursor positioning, insertion and deletion of lines within the page, and perhaps some automatic wraparound of long data strings from one line to the next. Support of such features is often distributed among the actual terminal, its system interface, and specialized application programs such as text editors or formatters.

3.4.2 Model Characteristics

The Conceptual Data Store contains a two dimensional array of storage cells for each paged display device of the negotiated terminal configuration. The array is called a page and is structured as an array of lines identical to those described in the discussion of Scroll Class services. A cursor is associated with each page to address the next character position into which data will be written.

An entry device may be configured to share access to a page. For such associations, a separate cursor may be negotiated, or the display cursor may be shared. Sharing of a page and/or its cursor also requires negotiation of the access control parameters governing use of these resources. An alternative configuration employs a separate basic or scrolling entry device and duplication of its data stream into the data stream for the page. This is effective when cursor movement and other paged services are not needed for entry purposes.

Each paged device is assigned an identifier to facilitate device addressing. A page is parameterized by its height, vertical and horizontal tab settings, and page overflow handling, together with the parameters for each line as described for scrolling devices. Some of these parameters may be interpreted differently in the context of Paged Class services. For example, horizontal tab settings are often constrained to apply uniformly to all lines of the display, and line overflow handling may allow for text wraparound to the beginning of the next line. Page overflow handling generally allows options similar to those for line overflow handling. However, paged operations may include additional options. Graphic rendition parameters may be associated with specific character positions rather than with the character data stored in the position. Such variations in capabilities may require more extensive use of the negotiation mechanisms than necessary for less sophisticated service classes.

3.4.3 Minimal Services

Establishment. Establishment services are described in the discussion of the generic services.

Negotiation. Negotiation services are described in the discussion of the generic services.

Termination. Termination services are described in the discussion of the generic services.

Data Transfer. The minimal services provided by the Paged Class include all minimal services of the Basic Class. Additionally, the following services are required for the data transfer services of the Paged Class:

a. Positioning of the cursor to the beginning of the next line (corresponds to the line completion indication for scrolling terminals)

b. Explicit indication of completion of modifications to a page

c. Erasure of the page and positioning of the cursor to the beginning of the first line in preparation for subsequent page operations

3.4.4 Additional Services

Data Transfer. Additional services may be implemented to allow the Paged Class to support the additional features of some paged-mode terminals. Use of these services is subject to negotiation.

a. Horizontal tabulation services

b. Vertical tabulation services

c. Explicit positioning of the cursor to the beginning of the first line (without erasure)

d. Absolute cursor positioning

e. Relative cursor movement

f. Dynamic selection of alternative character sets

g. Dynamic selection of graphic rendition parameters

h. Selective erasure of portions of the current line or page

i. Character insertion (existing characters within the current line are shifted right to create space for inserted characters)

j. Character deletion (complements insertion)

k. Line insertion (existing lines are shifted to create empty lines)

l. Line deletion (complements insertion)

3.5 FORMS CLASS VIRTUAL TERMINAL SERVICES

3.5.1 Introduction

Forms Class services support exchanges of data for defining, displaying, and utilizing forms for entry or display of structured data. A form is a secondary data structure associated with a display page like that described in the discussion of Paged Class services. A form consists of two parts, called the narrative and the record descriptor.

The record descriptor consists of a collection of fields. Each field has an identifier, a page position, a length, a data type, and attributes governing the display and manipulation of the data that it may contain. It is associated with the sequence of consecutive character positions beginning at its page position and limited by its length. Fields do not overlap in the page and usuallr do not cross line boundaries. The collection of data values associated with the fields is called a record. Field valuew are stored as character strings in the positions of the display page associated with the fields.

The aggregate of all non-field positions is available for storing the narrative of the form, which consists of text and other character strings intended to convey to a human user the nature of the field values. The non-field positions are manipulated using only the characteristics of the display page. Narrative may not be written into field positions.

The data stream format for forms operations is like that for paged and scrolling operations. It consists of sequences of character strings punctuated by control elements to indicate the intended interpretation of the data. However, additional types of control elements may occur to provide manipulation of the additional structure imposed on data for forms operations. Two main types of interaction are possible using forms terminals. One type is based on the aggregate exchange of data in units of whole forms or records. The other type accommodates interactive exchanges of data in units consisting of individual fields, narrative text, or field values.

Paged terminals may often be used for forms operations by means of software supports that superimpose form structure upon the display page. Similarly, forms terminals often operate as paged terminals when the interaction is restricted to the exchange of narrative text.

3.5.2 Model Characteristics

The Conceptual Data [tore contains a display page and an associated form for each forms display device of the negotiated terminal configuration. Depending on the access control capabilities negotiated, form definition and use may occur only in alternate stages of the data transfer phase or may be intermixed dynamically to support interactive dialogs.

In addition to the activity attribute, character positions of the display page have an attribute8indicating whether the position is associated with a field. Usually, only one application-entity is permitted to write into non-field positions during a specific dialog. This entity is called the form manager and others are called respondents. Such constraints are governed by access control.

Field attributes may affect the manipulation and display of data. Field attributes may include protection, entry requirements, visibility, transferability, display alteration, and content status. Protected fields may usually be written by the form manager but not by respondents. Other fields may require an entry by respondents. Visibility of a field determines whether an entered value is displayed. Nontransferable fields are not transferred to auxiliary devices, though such operations may be supported. Field values are often displayed with graphic attributes distinct from those for narrative text. Content status may indicate whether a field value has been modified by entry since the last exchange with a remote partner or if its value is undefined.

The data type of a field may constrain the values that may be entered into the field. Most types are subtypes of either string or number. Frequent qualifications of string types govern inclusion of lowercase alphabetic characters or other character subsets. Qualifications of number types control inclusion of a sign, the number of fractional digits permitted or required, and the ranges of magnitude allowed. Standardization of permissible field attributes and parameters requires further study.

3.5.3 Minimal Services

Establishment. Establishment services are described in the discussion of the generic services.

Negotiation. Negotiation services are described in the discussion of the generic services.

Termination. Termination services are described in the discussion of the generic services.

Data Transfer. Minimal services of the Forms Class include the minimal services of the Paged Class. However, these services are constrained to affect only non-field positions. Additionally, the following services are required for the data transfer services of the Forms Class:

a. Normal transfer of field values together with an indication of the field affected

b. Normal transfer of field definitions allowing for specification of all associated parameters

c. Deletion of all form and page data

d. Explicit indication of record update completion

e. Erasure of fields in preparation for subsequent entry or display of records

3.5.4 Additional Services

The following services may be implemented to allow the Forms Class to support the additional features of some terminals. Most of these are based on services adapted from the Paged Class.

a. Positioning to the next field (or next unprotected field)

b. Selective erasure of narrative text displayed in non-field positions

c. Dynamic modification of field attributes

d. Selective erasure of field values

e. Selective deletion of field definitions

f. Horizontal tabulation services

g. Vertical tabulation services

h. Positioning of the cursor to the beginning of the first line

i. Absolute cursor positioning

j. Relative cursor movement

k. Dynamic selection of alternate character sets

l. Dynamic selection of graphic rendition parameters

4. UPPER LAYER SERVICE/INTERFACE SPECIFICATION

TO BE SUPPLIED

5. SERVICES REQUIRED FROM THE LOWER LAYERS

The virtual terminal service requires certain services provided by the session layer. The following list of required services is based on the current ISO session service specification [ISO, 1981c]. The services are organized according to the phases of operation in which they will be used. General services can be used in any phase.

This list is a conservative prediction of the services required from the session layer. It is likely that the following facilities will not be required after refinement of the design:

a. Quarantine Service

b. Token Management Service

c. Synchronization Point Service

d. Resynchronization Service

5.1 ESTABLISHMENT PHASE SESSION SERVICES

Connection Establishment Service. This service establishes a connection to a given foreign host. A foreign host can accept or reject any connection attempt, and the service places no architectural restriction on the number of connections that may be concurrently established. Connections are established according to quality of service parameters including transit delay, setup delay, and error rate.

5.2 NEGOTIATION PHASE SESSION SERVICES

Reselection Service. This service allows the presentation layer to select new values for each session parameter and the availability of certain services (within the range defined during session establishment phase).

Token Management Sebvice. The token management service provides the session service user with the following services:

a. Give tokens - allows the session user to surrender one or more tokens

b. Request tokens - allows a session user to bid for ownership of specific tokens

c. Release tokens - allows a session user to indicate that the implied service is no longer required

5.3 DATA TRANSFER PHASE SESSION SERVICES

Data Transfer Service. This service provides for the transparent transmission of data as a stream of octets. Thus, it does not restrict the content, format, or encoding of commands or data. The transmission is in order, and without loss or duplication of data unless explicitly reported.

Expedited Service. The expedited data exchange service allows both service users to exchange expedited session service data-units. The exchange of expedited-service-data-units is not subject to the right to send normal data, flow control of normal data, or the quarantine service. An expedited-service-data-unit will not be delivered later than a normal service data-unit or an expedited-service-data-unit that is sent subsequent to the expedited-service-data-unit.

Quarantine Service. The quarantine service allows the sending session service user to designate that some number of session service data units are only meaningful when all of them are available. The receiving session entity does not release these session service data units to the receiving session user until the end of the quarantine unit is delimited by the sending session-entity.

Two-Way Simultaneous Interaction Service. This service allows both ends of a connection to concurrently send and receive data.

5.4 TERMINATION PHASE SESSION SERVICES

Orderly Release Service. This service allows either service user to request the termination of the connection without loss of data. The user invoking this service is provided with the means to determine whether the other user accepts or refuses the termination. If it is refused, the connection is not terminated, and no data is lost.

P-Abort Service. This service is the means by which the service provider may terminate the connection for reasons internal to the service provider, when the underlying transport service is unavailable, or when the quality of service falls below specified levels. Use of this service may cause loss of data.

5.5 GENERAL SESSION SERVICES

Synchronization Point Service. This service allows a user to request a major synchronization point. Such a request completely separates all communication before the synchronization point from communication after it. The session service assigns a serial number to the synchronization point for possible later use by the service user in requesting the resynchronization service. This service may be used by the THP-3 to separate and distinguish the phases of virtual terminal service operation.

Resynchronization Service. This service resets the session connection to a defined state and purges all data not yet delivered. The service user can reset the connection to a previous synchronization point by specifying its

serial number.

Exception Reporting Service. This service fotifies the users of unexpected situations not covered by the other services. Typically, these situations are due to service errors or malfunctions.

6. LOWER LAYER SERVICE/INTERFACE SPECIFICATION

TO BE SUPPLIED

7. PROTOCOL ENTITY SPECIFICATION

TO BE SUPPLIED

8. EXECUTION ENVIRONMENT SPECIFICATION

TO BE SUPPLIED

9. GLOSSARY

TO BE SUPPLIED

BIBLIOGRAPHY

1. ARPA, Telnet Protocol Specification, NIC 18639, 1973. Also in ARPANET Protocol Handbook, 1973.

2. CSC, Terminal-to-Host Protocol, Computer Science Corporation, 1979.

3. Day, J., Telnet Data Entry Terminal Option, ARPANET RFC 731, 1977.

4. Day, J., Terminal Protocols, IEEE Transactions on Communications, Vol. COM-28 No. 4, April 1980.

5. ECMA, Generic Virtual Terminal Model and Service Definition, 4th Draft, ECMA/TC23/81/53, 1981a.

6. ECMA, Virtual Terminal Protocol for the Basic Class, 10th Draft, ECMA/TC23/81/72, 1981b.

7. IFIP, Proposal for a standard virtual terminal protocol, INWG Note #91, 1978. Also in Proc. Symp. Computer Network Protocols, Liege, Belgium, 1978.

8. ISO, Data Processing - Open Sytems Interconnection - Basic Reference Model, ISO DP 7498, 1981a.

9. ISO, Generic Virtual Terminal Model and Service Definition, ISO TC97/SC16/N713, 1981b.

10. ISO, Connection-Oriented Session Service Specification, ISO TC97/SC16/N700, 1981c.

11. Nabielsky, J. and A. Skelton, A Virtual Terminal Management Model, ARPANET RFC 782, 1981.

12. Postel, J., L. Garlick, and R. Rom, Terminal-to-Host Protocol Specification, Stanford Research Institute, 1976.

13. Schicker, P. and A. Duenki, Definition and Virtual Terminal Protocol, EIN/EHR/76/019a; INWG Note #51, 1977.

14. SDC, DCEC Protocols Standardization Program Protocol Specification Report, SDC TM-7038/204/00, 1981.

15. SYTEK, DoD Architectural Model, SYTEK, Inc., 1980.